REMARKS

Claims 1-27 are pending in the present application. Claims 1-27 are amended. Claims 1, 11, 12, and 20 are amended to recite "determining if a current panel is a custom panel; if the current panel is a custom panel, retrieving a set of navigation objects from the current panel; determining, for each navigation object in the set, if an event associated with each navigation object is one of a linking event and a non-linking event, wherein the linking event links the current panel to a target panel; if the event is a linking event, assigning the current panel as the target panel; creating a set of data objects for display in the target panel, wherein each data object within the set of data objects is associated with a data type; and creating a set of navigation objects for display in the target panel, wherein each navigation object within the set of navigation objects is associated with an event." These features are supported in Figure 9, Figure 10, and page 15, line 13 to page 17, line 5 of the current specification.

Claims 2, 13, and 21 are amended to recite "wherein properties of the set of data objects and properties of the set of navigation objects for the current and target panel are stored in a plurality of entries in a database, and wherein each entry includes a set of name-value pairs for the properties of the set of data objects and a set of name-value pairs for the properties of the set of navigation objects for a different panel." These features are supported at least on page 13, lines 6-15, and in Figure 4 of the current specification.

Claims 3, 15, and 22 are amended to recite "wherein each of the set of name-value pairs for the properties of the set of navigation objects includes a name of each navigation object and a value of a target panel to which each navigation object is linked." These features are supported at least on page 13, lines 20-22 and in Figure 7 of the current specification.

Claims 4, 16, and 23 are amended to recite "wherein each of the set of name-value pairs for the properties of the set of data objects includes a value of a data type associated with each data object and a name of each data object." These features are supported at least on page 13, lines 15-19 and in Figure 7 of the current specification.

Claims 5, 17, and 24 are amended to recite "if the current panel is a custom panel, attaching a header to the name of each data object of the current panel, wherein the header uniquely identifies a data object across multiple panels; and storing the set of

navigation objects and a set of data objects for the current panel to a set of name-value pairs in an entry of the database." These features are supported at least on page 14, line 29 to page 15, line 9 of the current specification.

Claims 6, 18, and 25 are amended to recite "if the event is a non-linking event and is an OK event, retrieving user input data of each data object in the set of data objects from the current panel and linked panels; and storing the user input data to the database." These features are supported at least on page 17, lines 4-9 of the current specification.

Claims 7, 19, and 26 are amended to recite "if the event is a non-linking event and is a Cancel event, closing the current panel; and discarding the user input data." These features are supported at least on page 14, line 15-20 of the current specification.

Claims 8, 10, 20, and 27 are amended to recite "if the event is a non-linking event and the event performs a task, creating a set of information objects for display in the target panel." These features are supported at least on page 12, lines 1-8 of the current specification.

Claim 9 is amended to recite "a linked target panel for use in a graphical user interface, the panel comprising: a set of data objects displayed in the target panel; a set of navigation objects displayed in the target panel; wherein the set of data objects are created from properties of the set of data objects stored in a database, wherein each property includes a name-value pair with a value of a data type for each data object and a name of each data object name; and wherein the set of navigation objects are created from properties of the set of navigation objects stored in the database, wherein each navigation object in the set is associated with an event, and wherein each property includes a name-value pair with a name of each navigation object name and a value of a target panel to which each navigation object is linked." These features are supported at least on page 13, lines 15-22 and in Figure 7 of the current specification.

Reconsideration in view of the above amendments to claims and the following remarks is respectfully requested.

I. 35 U.S.C. § 101, Claims 20-27

The Office Action rejects claims 20-27 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. The Office Action states that the computer

program product in a computer readable medium is considered non-statutory because the claims contain no reference directed toward the action of computer executing the computer program product. By this Response, independent claim 20, from which claims 21-27 depend, is amended to recite a computer program product in a computer readable medium executed by a computing device for linking panels in a graphical user interface. The amendments are made to include an action of a computing device executing the computer program product. Thus, Applicants respectfully submit the withdrawal of the rejection of claims 20-27 under 35 U.S.C. § 101.

II. Objection to Claims 8, 19, and 27

The Office Action stated that claims 8, 19, and 27 were objected to as being dependent upon rejected base claims 1, 12, and 20. The Office Action also stated that the claims do not further limit claims 1 and 12 respectively because the information objects in claims 8 and 19 contain no distinction from the data object of claims 1 and 12. By this Response, claims 8, 19, and 27 are amended to recite creating a set of information objects for display in the target panel if the event is a non-linking event and the event performs a task. As described page 12 of the current specification, a button may simply perform a task, which does not bring up an additional panel. This kind of button does not participate in any navigation and is a "non–link" button. Thus, a set of information objects as recited in claims 8, 19, and 27 are objects that performs a task. This is different from a set of data objects as recited in claims 1, 12 and 20, which simply displays data. Therefore, Applicants respectfully submit that claims 8, 19, and 27 include limitations not found in claims 1, 12 and 20 and thus request the objections to claims 8, 19, and 27 to be withdrawn.

III. 35 U.S.C. § 102(b), Alleged Anticipation, Claims 1-3, and 6-14

The Office Action rejects claims 1-3 and 6-14 under 35 U.S.C. § 102(b) as being allegedly anticipated by Selby et al. (U.S. 5,555,365). This rejection is respectfully traversed.

As to claims 1-3 and 6-14, the Office Action states:

> As per claims 1, 12, and 20, Selby teaches: creating a set of data objects for a panel (line 64 of column 4 through line 12 of column 5); wherein each data object is associated with a data type (line 64 of column 4 through line 1 of column 5; data types are also inherent for data objects); creating a set of navigation objects for the panel (lines 39-53 of column 2 and Fig. 4); wherein each navigation object is associated with an event, a target object associated with the event, and a target object action associated with the target object in which the target object action is applied to the target object (lines 42-53 of column 2, line 64 of column 7 through line 2 of column 8, and Fig. 8).

Office Action dated November 15, 2004, page 3.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 21 U.S.P.Q.2d 1031, 1034 (Fed Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). Applicants respectfully submit that Selby does not teach every element of the claimed invention arranged as they are in claims 1 and 12. Specifically, Selby does not teach determining if a current panel is a custom panel; if the current panel is a custom panel, retrieving a set of navigation objects from the current panel; determining, for each navigation object in the set, if an event associated with each navigation object is one of a linking event and a non-linking event, wherein the linking event links the current panel to a target panel; if the event is a linking event, assigning the current panel as the target panel; creating a set of data objects for display in the target panel; and creating a set of navigator objects for display in the target panel.

Amended independent claim 1, which is representative of claims 11, 12, and 20 with regard to similarly recited subject matter, now recites:

> 1. A method in a data processing system for linking panels in a graphical user interface, the method comprising:
>
> determining if a current panel is a custom panel;
>
> if the current panel is a custom panel, retrieving a set of navigation objects from the current panel;

determining, for each navigation object in the set, if an event associated with each navigation object is one of a linking event and a non-linking event, wherein the linking event links the current panel to a target panel;

if the event is a linking event, assigning the current panel as the target panel;

creating a set of data objects for display in the target panel, wherein each data object within the set of data objects is associated with a data type; and

creating a set of navigation objects for display in the target panel, wherein each navigation object within the set of navigation objects is associated with an event. (emphasis added)

Selby does not teach the features emphasized above. As discussed in the Abstract, Selby teaches an object-oriented environment that includes a plurality of graphic user interface objects and a plurality of application objects. Each graphic user object has a unique identifier and a selected attribute and each application object has a preselected response. The system of Selby identifies graphic user interface objects associated with an application object and creates a table specifying relationships between the graphic user interface objects and between each graphic user interface object and the application object. Each time the application object is initialized, the table is used to specify objects within the application object.

Selby first does not teach determining if a current panel is a custom panel or if the current panel is a custom panel, retrieving a set of navigation objects from the current panel. In Figure 4, Selby teaches a panel that includes a number of controls. At column 5, lines 15-25, Selby also teaches a method for dynamic binding between resource identifiers of the controls on panels and object attributes. Selby binds the resource identifiers with application object attributes. In addition, at column 2, lines 10-15, Selby teaches that a GUI object is an object visible to a user and application object is an object that is not visible. The binding is created between a GUI object and an application object to allow interaction between a user and the application object using a GUI.

Thus, Selby merely teaches how to bind GUI object or controls visible in a panel to application objects that are not visible to the user, in order to allow interaction between the user and the application object. Selby does not teach determining if a current panel is a custom panel or retrieving a set of navigation objects from the current panel if the

current panel is a custom panel. To the contrary, as described in the above sections, Selby is only concerned with binding controls that are visible in a panel. Selby is not concerned with whether the panel itself is a custom panel or retrieving a set of navigation objects from the current panel if the panel is a custom panel. Therefore, Selby fails to teach determining if a current panel is a custom panel or if the current panel is a custom panel, retrieving a set of navigation objects from the current panel, as recited in claims 1 and 12 of the present invention.

In addition, Selby does not teach determining, for each navigation object in the set, if an event associated with each navigation object is one of a linking event and a non-linking event, wherein the linking event links the current panel to a target panel. Figure 4, which illustrates a diagram of a panel, and Figure 5 of Selby, which illustrates a panel resolution table, are shown below:

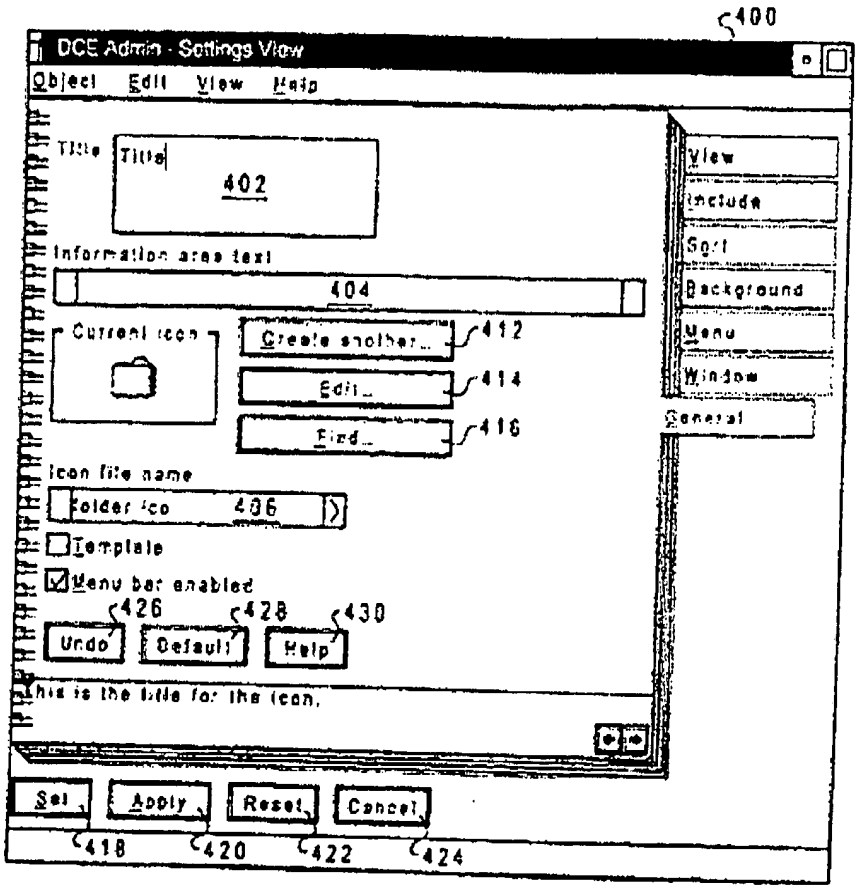


Fig. 4

As shown in **Figure 4**, while Selby teaches a set of navigation objects, such as create another **412**, edit **414**, and find **416**, Selby does not teach determining for each navigation object, if an event associated with each navigation object is one of a linking event and a non-linking event, wherein the linking event links the current panel to a target panel.

At column 6, lines 12-22, Selby teaches that for each control in panel **400** defined in panel resolution table **500**, dependencies are checked for each control. If the dependency is not equal to NULL, it checks the attribute name in the control that is dependent on and enables or disables the control accordingly. Selby also teaches, at column 6, lines 23-37, that push buttons create another **412**, edit **414**, and find **416** are depending on icon file name **406** as can be seen with reference to fields under Depend On ID column in table **500**. Thus, in this case, controls **412**, **414**, and **416** only link to other controls within the panel, such as control **406**, or application objects, which are invisible. None of the controls in **Figure 4** links to anything visible outside of panel **400**.

500

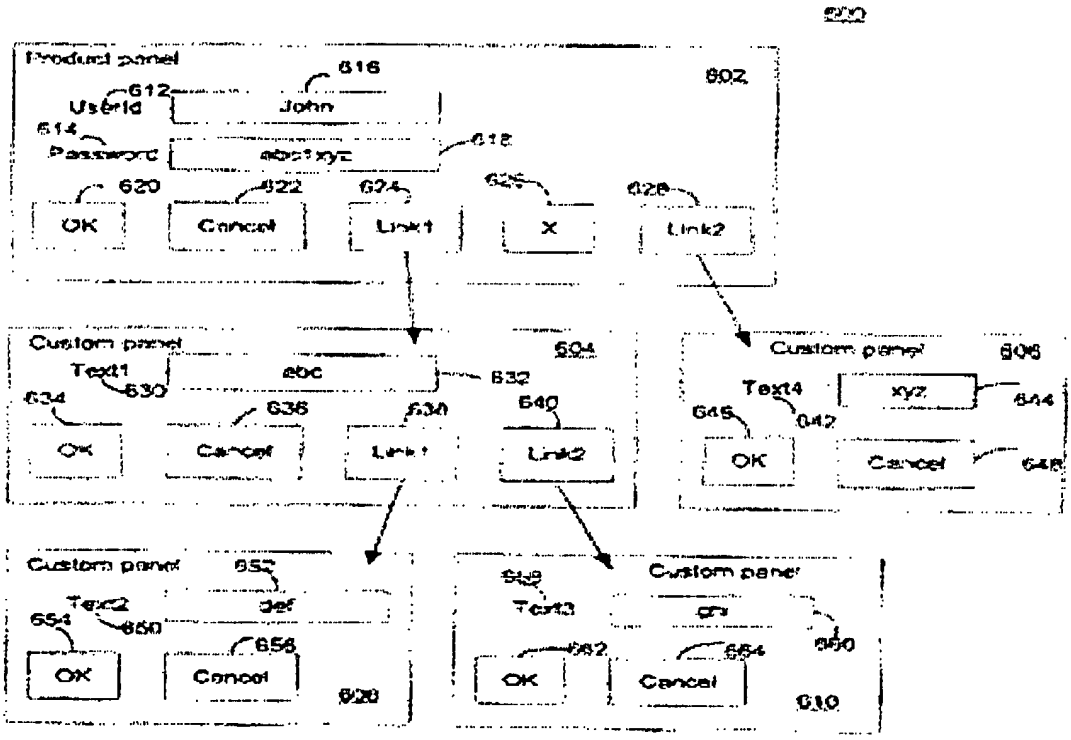
Resource ID	Control Type	Attribute Name	Depend On ID	Dependency Function
rID1	EF	Title	0	
rID2	EF	Info_area_text	0	
rID3	PB	NULL	rID6	
rID4	PB	NULL	rID6	
rID5	PB	NULL	rID5	
rID6	EF	Icon_File_Name	0	IconFileEF
rID7	CB	Template_Flag	0	
rID8	CB	Menubar_Flag	0	

Fig. 5

At column 6, lines 37-54, Selby teaches a dependency function in **Figure 5**, which provides the ability to customize interaction between GUI object and application object for cases where the generic mechanism is insufficient. Thus, while Selby teaches a dependency function that adds customization to the GUI objects within the panel, the

dependency function only links the controls to non-visible application objects. The dependency function does not link any control in panel **400** to another panel.

In addition, Selby does not mention anything about a target panel or a linking event that links a current panel to a target panel, since Selby is only concerned with linking controls to other controls within a panel or to application objects, not to another panel. Selby's teachings of linking controls in a panel is different from the presently claimed invention, in that the controls in Selby's teachings are linked to either the controls of the same panel or application objects that are invisible. This is contrary to the present invention, as illustrated in **Figure 6** of the current specification shown below, where link buttons, such as link1 **624**, of current panel **602** associates with a link event that links current panel **602** to target panel **604**. Thus, Selby does not teach determining, for each navigation object in the set, if an event associated with each navigation object is one of a linking event and a non-linking event, wherein the linking event links the current panel to a target panel, since Selby does not teach any event of a control that links the current panel to a target panel.



Furthermore, Selby does not teach if the event is a linking event, assigning the current panel as the target panel. As discussed above, Selby does not teach a linking event that links a current panel to a target panel. Therefore, Selby would not teach if the event is a linking event, assigning the current panel as the target panel. In addition, Selby is only concerned with linking controls to application objects or to other controls within the panel. Selby is not concerned with linking controls to a target panel. Therefore, Selby does not and would not teach if an event is a linking event, assigning the current panel as the target panel, because assigning the current panel as the target panel would not link controls of a panel to either other controls of the same panel or to application objects.

Moreover, Selby fails to teach creating a set of data objects for display in the target panel or creating a set of navigation objects for display in the target panel. The Office Action alleges that Selby teaches these features at column 4, line 64 to column 5, line 12, column 2, lines 42-53, and column 7, line 64 to column 8, line 2, which read as follows:

> Objects are grouped into classes of related objects. The class description contains information relevant to all objects in a class, including a description of instance variables maintained by each of the objects and the available object methods. An object instance is created (or "instantiated") based on that information and has the properties defined in the object class. For example, the object class DOG can include the instance variables "dog_type" and "dog_name" and a "bark" method implementing the response to a bark message. An instance of a dog, e.g. ROVER, will maintain the type and name instance variables for itself and will respond to bark message.
>
> Abstract classes are used to describe the interfaces and methods expected to be used by a class without providing detail on the implementation of those methods. Abstract classes are useful in framework where the implementation details are to be left to the implementor. Concrete classes are created as subclasses of abstract classes and implement those classes.

(Column 4, line 64 to column 5, line 12)

> The present invention provides a data processing system having an object oriented environment, wherein the object oriented environment includes a plurality of graphic user interface objects and a plurality of application objects, each graphic user object having a unique identifier and a selected attribute and each application object having a presselected response. Graphic user interface objects associated with an application

> object are identified. A table is created specifying relationships between the graphic user interface objects and each graphic user interface object and the application object. Each time the application object is initialized, the table is used to specify objects within the application object.
> (Column 2, lines 42-53)

> With reference to **Figure 8**, a flowchart of a process for managing the reading of information from a panel is depicted in accordance with a preferred embodiment of the present invention. The process retrieves information from the controls and sets values in the associated data objects' attributes when a user clicks on a pushbutton which invokes a method which will read the panel, the user closes the window and the panel must be read, and a process decides to read the panel.
> (Column 7, line 64 to column 8, line 2)

In the first section, Selby teaches that application objects are objects that include instance variables and associated methods. In the second section, Selby merely teaches how to associate graphic user object to application object using a unique resource identifier. In the third section, Selby merely teaches how to read information from the controls of a panel and sets values of application object attributes when a user clicks on a push button. However, in none of the above sections, or any other section, does Selby mention anything about creating a set of data objects for display in a target panel or creating a set of navigation objects for display in a target panel. As described above, Selby is only concerned with binding controls in a single panel with application objects. Selby does not teach any event that links a current panel to a target panel. Therefore, Selby does not and would not teach creating a set of data objects or navigation objects for display in the target panel.

In view of the above, Applicants respectfully submit that Selby does not teach each and every feature of claims 1, 11, 12, and 20. At least by virtue of their dependency on claims 1 and 12 respectively, Selby does not teach or suggest the features of dependent claims 2, 3, and 6-14. Accordingly, Applicants respectfully request the withdrawal of the rejection of claims 1-3 and 6-14 under 35 U.S.C. § 102(b).

In addition, Selby does not teach the specific features of claims 2, 3, 6-14. For example, with regard to amended dependent claim 2, which is representative of claims 13 and 21 with regard to similarly recited subject matter, Selby does not teach wherein properties of the set of data objects and properties of the set of navigation objects for the

current and target panel are stored in a plurality of entries in a database, and wherein each entry includes a set of name-value pairs for the properties of the set of data objects and a set of name-value pairs for the properties of the set of navigation objects for a different panel.

While Selby teaches in **Figure 5** a panel resolution table with a plurality of entries, each entry of the panel resolution table does not include a set of name-value pairs for properties of the set of data objects and properties of the navigation objects. At column 5, lines 55-66, Selby teaches a Control Resource ID which uniquely identifies a control in a panel, a Control Type identifying a type of control, an Attribute Name identifying a name given to the instance data of the object associated with the control, a Depend On ID identifying a Resource ID of the control that this control depends on, and dependency function, which may be customized and invoked every time the panel is displayed, read, or changed.

However, each of the attribute values is arranged as individual columns within table **500**. The attribute values are not arranged as a name-value pair, as illustrated in **Figure 7** of the current specification below:

Panel	Data 1
Custom panel CP1 702	data property: {JTextField=Text1} navigation buttons property: {Link1=CP2, Link2=CP3}
Custom panel CP2 704	data property: {JTextField=Text2} navigation buttons property: { }
Custom panel CP3 706	data property: {JTextField=Text3} navigation buttons property: { }
Custom panel CP4 708	data property: {JTextField=Text4} navigation buttons property: { }

As shown in **Figure 7** of the current specification, for each custom panel, a set of name-value pairs for properties of the set of data objects are stored as data property, while a set of name-value pairs for properties of the set of navigation objects are stored as

navigation property. Selby does not teach such sets of name-value pairs. To the contrary, Selby teaches storing each of the attribute values in a separate column as illustrated in table **500** of **Figure 5**. Therefore, Selby does not teach the features of claims 2, 13 and 21 of the present invention.

With regard to amended dependent claim 3, which is representative of claims 14 and 22 with regard to similarly recited subject matter, Selby does not teach that each of the set of name-value pairs for the properties of the set of navigation objects includes a name of each navigation object and a value of a target panel to which each navigation object is linked. As described above in claim 2, Selby does not teach a set of name-value pairs for properties of the set of navigation objects. Therefore, Selby would not teach that each of the set of name-value pairs for the properties of the set of navigation objects includes a name of each navigation object and a value of a target panel to which each navigation object is linked.

In addition, Selby does not teach a value of a target panel to which each navigation object is linked. As discussed in arguments presented above for claims 1 and 12, Selby teaches linking controls of a panel to either other controls of the panel or application objects that are invisible. This is different from the presently claimed invention, as illustrated in **Figure 7** of the current specification reproduced above, where the navigation property value pairs include a name of a navigation object, such as Link1, and a target panel to which each navigation object is linked, for example, CP2. Therefore, Selby fails to teach the feature of claims 3, 14, and 22 of the present invention.

With regard to amended dependent claim 6, which is representative of claims 17 and 25 with regard to similarly recited subject matter, Selby does not teach if the event is a non-linking event and is an OK event, retrieving user input data of each data object in the set of data objects from the current panel and linked panels, and storing the user input data to the database. As discussed in arguments presented for claims 1 and 12, Selby does not teach determining if an event associated with the navigation object is one of a linking and non-linking event, therefore, Selby would not teach if the event is a non-linking event and is an OK event, retrieving user input data of each data object in the set of data objects from the current panel and linked panels. In addition, Selby does not teach retrieving user input data of each data object in the set of data objects from the

current and linked panels, since Selby does not link controls of one panel to another panel. Selby merely links controls of a panel to other controls of the same panel or application objects that are invisible. Therefore, Selby does not teach the features of claims 6, 17, and 25 of the present invention.

With regard to amended dependent claim 7, which is representative of claims 18 and 26 with regard to similarly recited subject matter, Selby does not teach if the event is a non-linking event and is a Cancel event, closing the current panel and discarding the user input data. As discussed in arguments presented for claims 1 and 12, Selby does not teach determining if an event associated with the navigation object is one of a linking and non-linking event, therefore, Selby would not teach if the event is a non-linking event and is a Cancel event, closing the current panel. Therefore, Selby does not teach the features of claims 7, 18, and 26 of the present invention.

With regard to amended dependent claim 8, which is representative of claims 19 and 27 with regard to similarly recited subject matter, Selby does not teach if the event is a non-linking event and the event performs a task, creating a set of information objects for display in the target panel. The Office Action alleges that Selby teaches these features at column 4, lines 64 to column 5, line 12, which is reproduced above. However, in this section, Selby merely teaches that application objects are objects that include instance variables and associated methods. As described above, application objects are invisible objects. They are not a set of information objects for display in a target panel. Therefore, Selby also fails to teach the features of claims 8, 19, and 27 of the present invention.

With regard to amended dependent claim 9, which is representative of claims 20 and 28 with regard to similarly recited subject matter, now recites:

> 9. A linked target panel for use in a graphical user interface, the panel comprising:
>
> a set of data objects displayed in the target panel;
>
> a set of navigation objects displayed in the target panel;
>
> wherein the set of data objects are created from properties of the set of data objects stored in a database, wherein each property includes a name-value pair with a value of a data type for each data object and a name of each data object name; and
>
> wherein the set of navigation objects are created from properties of the set of navigation objects stored in the database, wherein each

navigation object in the set is associated with an event, and wherein each property includes a name-value pair with a name of each navigation object name and a value of a target panel to which each navigation object is linked. (emphasis added)

Selby does not teach a set of data objects that are created from properties of the set of data objects stored in a database, wherein each property includes a name-value pair with a value of a data type for each data object and a name of each data object name. In **Figure 5** reproduced above, Selby merely teaches storing attribute values of a control in a panel resolution table. However, according to Selby, each attribute value of a control, for example, attribute name and control type, is stored in individual columns of the table. Selby does not teach storing the attributes of data objects in a name-value pair with a value of a data type for each data object and a name of each data object name. Selby also does not teach storing each attribute in a name-value pair with a name of each navigation object and a value of a target panel to which each navigation object is linked. Therefore Selby does not teach the features of claims 9, 20, and 28 of the present invention.

With regard to amended dependent claim 10, Selby does not teach a set of information objects for display in the target panel, wherein the set of information objects are created to performs a set of tasks. The Office Action alleges that Selby teaches a set of information objects at column 4, line 64 to column 5, line 12, which is reproduced above. However, as discussed in arguments presented for claims 8, 19, and 27, Selby merely teaches that application objects are objects that include instance variables and associated methods. As described by Selby, application objects are invisible objects. They are not a set of information objects for display in a target panel. Therefore, Selby also fails to teach the features of claim 10 of the present invention.

In view of the above, in addition to their dependency on independent claims 1 and 12, Applicants respectfully submit that Selby also fails to teach the specific features of claims 2, 3, and 6-14 of the present invention. Accordingly, Applicants respectfully request the withdrawal of rejection of claims 1-3 and 6-14 under 35 U.S.C. § 102(b).

IV. 35 U.S.C. § 103(a), Alleged Obviousness, Claims 4-5, 15-16, and 23-24

The Office Action rejects claims 4-5, 15-16, and 23-24 under 35 U.S.C. § 103(a) as being allegedly unpatentable over Selby as applied to claims 1 and 12 respectively above, and further in view of Official Notice. This rejection is respectfully traversed.

As to claims 4-5, 15-16, and 23-24, the Office Action states:

> As per claims 4, 15, and 23, Official Notice is taken of the event being a right click. Right clicks causing an event has been well known in the art since the first two-button mouse was implemented. It is for this reason that it would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to make right clicks on a mouse produce an event in the system as taught by Selby.

Office Action dated November 15, 2004, page 5.

Amended dependent claim 4, which is representative of claims 15 and 23 with regard to similarly recited subject matter, now recites:

> 4. The method of claim 3, wherein each of the set of name-value pairs for the properties of the set of data objects includes a value of a data type associated with each data object and a name of each data object

As discussed in arguments presented for claims 2, 13, and 21, Selby does not teach a set of name-value pairs for properties of the set of data objects. To the contrary, Selby teaches storing each attribute value in a separate column within a panel resolution table. Therefore, Selby would not teach a set of name-value pairs, let alone including a value of a data type associated with each data object and a name of each data object in each of the set of name-value pairs for properties of the set of data objects. Therefore, Selby does not teach the features of claims 4, 15, and 23 of the present invention.

Amended dependent claim 5, which is representative of claims 16 and 24 with regard to similarly recited subject matter, now recites:

> 5. The method of claim 4, wherein further comprising:
> if the current panel is a custom panel, attaching a header to the name of each data object of the current panel, wherein the header uniquely identifies a data object across multiple panels; and
> storing the set of navigation objects and a set of data objects for the current panel to a set of name-value pairs in an entry of the database. (emphasis added)

Selby does not teach attaching a header to the data object name of each data object of the current panel, wherein the header uniquely identifies a data object across multiple panels. As discussed in arguments presented in claims 1, 12, and 20, Selby is only concerned with linking controls to other controls of the same panel or to application objects that are invisible. Selby is not concerned with linking a current panel to a target panel. Nowhere in the reference does Selby teach or suggest anything about attaching a header to a data object name, let alone a header that uniquely identifies a data object across multiple panels. Since Selby is not interested in linking one panel to another panel, there is no need for Selby to attach a header that uniquely identifies the data object across multiple panels. Therefore, Selby fails to teach or suggest attaching a header to the data object name of each data object, wherein the header uniquely identifies a data object across multiple panels. In addition, Selby would not teach such features, because Selby is only concerned with a single panel. Therefore, Selby does not teach or suggest the features of claims 5, 16, and 24 of the present invention.

In view of the above, Applicants respectfully submit that Selby also fails to teach or suggest the features of claims 4-5, 15-16, and 23-24 of the present invention. Accordingly, Applicants respectfully request the withdrawal of rejection of claims 4-5, 15-16, and 23-24 under 35 U.S.C. § 103(a).

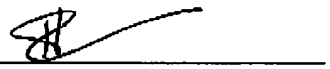
V. Conclusion

It is respectfully urged that the subject application is patentable over Selby and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: Feb 15, 2005

Respectfully submitted,



Wing Yan Mok
Reg. No. 56,237
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Agent for Applicants